하드디스크 없는 클라이언트 PC를 위한 네트워크 컴퓨팅 시스템의 설계 및 구현

정연기[†] · 이광진[††]

요 약

DL-NCS (Network Computing System for Diskless Client PCs)에서는 Windows NT에서 제공하는 기능을 최대한 이용하여 윈도우 9x setup 디스켓, RPL 프로파일의 생성 방안 등을 제시하고, 원격부팅을 위한 RPL 프로토콜을 제안한다. 이렇게 하여 하드디스크 없는 클라이언트가 서버측 하드디스크에 저장된 윈도우 9x를 원격부팅할 수 있으며, 서버측 주변장치 및 응용 프로그램들을 공유할 수 있게 구현한다. 윈도우 9x를 DL-NCS 서버로부터 부팅하기 위해 25대의 클라이언트에서 동시에 PC의 전원 스위치를 켠 결과, 속도면에서도 독립된 PC에서 부팅할 때와 비슷한 시간 내에 부팅이 가능했다. 다른 응용프로그램에 대해서도 같은 실험을 실시해 본 결과, 우수한 네트워크 환경을 제공해 줌을 알 수 있었다. 본 DL-NCS를 활용하면, 주변장치 설치비용을 절감할 수 있고, 486SX/DX와 같은 낮은 성능의 PC를 재활용할 수 있으며, 소프트웨어 구입비용을 절감할 수 있다.

Design and Implementation of a Network Computing System for Diskless Client PCs

Chung, Youn-Ky[†] and Lee, Kowang-Jin[††]

ABSTRACT

With fully harnessing the functions provided by Windows NT, we suggest the generating method of RPL profile and the RPL protocol for remote booting on DL-NCS(Network Computing System for Diskless client PCs). This system can offer clients, without having hard disk, boot remotely Window 9x stored in server's hard disk and can share peripheral devices and application programs in the server side. For booting Window 9x from DL-NCS, twenty-five clients' PCs were set for simultaneous on and finally the PCs were booted in similar time interval compared to that from a desktop PC having local disk. This system was also experimented to the application programs. And this system can yield excellent networking environments. In sum, with using this DL-NCS, we can reduce the installing costs of peripheral device, recycle the lower grade PCs such as 486SX/DX, and save the budgets for software purchase.

1. 서 론

기업 정보 인프라의 유지비용을 줄이려는 고객의 요구를 충족시키기 위해, 오라클(Oracle), 어콘(Arcon), 선(Sun), IBM 등의 유수 기업들이 NC(Network Computer)를 개발하였다. 이에 자극을 받아 PC 업계의 두 리더인 마이크로소프트와 인텔은 1996년에 NetPC 개념을 발표했다[1,2].

IBM이 NC로 전환할 수 있도록 발표한 워크스페이스 온 디맨드(WSOD, WorkSpace On-Demand)를 이용하면, 기존의 PC를 위해 운영체제와 애플리케이션을 지원하는데 드는 비용을 크게 줄일 수 있다[3]. WSOD는 OS/2 기반의 애플리케이션 관리 툴이기 때문에, 단순화된 애플리케이션 분산 기능을 제공해 기존의 PC를 NC로 이전하려는 기업에게 유용한

† 경일대학교 컴퓨터공학과

†† 경일대학교 산업대학원 컴퓨터공학과

제품이다. 그러나, WSOD는 하드웨어 및 소프트웨어 요구사항이 까다롭고 마이크로소프트 윈도우 95/98/NT를 지원하지 못하는 단점이 있다[4]. 이와 같이 NC는 PC와 호환되지 않고, NetPC는 초기 구입 비용 면에서 볼 때 전통적인 PC보다 결코 저렴하지 않기 때문에, 오늘날까지 우리 나라에서는 보편화되지 못한 실정이다.

본 논문에서는 IBM이 개발한 WSOD와는 달리 PC/Windows NT를 서버로 두고, 각 클라이언트는 하드디스크(HDD)와 CD ROM 드라이브가 없는 씬 클라이언트(thin client)로 구현하였다. 클라이언트 PC의 LAN 카드에 부트(boot) ROM을 내장하여, 각 클라이언트 PC는 서버측 하드디스크로부터 윈도우 9x를 원격 부팅할 수 있게 하였고, 서버측 하드디스크/CD ROM과 같은 하드웨어 자원과 각종 애플리케이션들을 공유할 수 있게 하였다.

국내에서는 아직도 많은 중소기업체, 컴퓨터학원, 정보고등학교 등에서 486SX/DX나 펜티엄 100MHz 이하급의 낮은 성능을 가진 PC를 사용하고 있기 때문에, 윈도우 9x를 설치하여 사용하기엔 하드디스크의 용량이 부족하고, 하드웨어적 특성으로 인해 큰 용량의 하드디스크를 설치할 수도 없다. 본 논문에서 구현한 하드디스크 없는 네트워크 컴퓨팅 시스템(DL-NCS, Network Computing System for diskless client PCs)을 중소기업체, 초중고등학교, 컴퓨터학원 등에 설치하여 활용해 본 결과, 윈도우 9x 환경에서 서버측에 설치된 대부분의 애플리케이션을 공유하여 사용할 수 있었다.

서론에 이어 2장에서는 관련된 연구로서 WSOD에 대해 분석하고, 3장에서는 개발된 원격부팅 시스템의 구성과 프로토콜에 대해 분석한다. 4장에서는 구현 결과에 대해 분석하고 5장에서 결론을 맺는다.

2. 관련 연구

PC의 기능을 단순화해서, 컴퓨터 통신망의 각종 서버에 들어 있는 운영체제나 응용소프트웨어·멀티미디어 정보 등을 자유롭게 불러다 쓸 수 있게 한 통신전용 단말기를 NC라고 한다. NC는 IP 기반의 네트워크에 접속하여 다른 네트워크 노드와 네트워크 컨텐트를 주고 받는 장치이다[2]. IBM이 발표한 WSOD 1.0은 이렇게 NC로의 이전을 고민하고 있는 네트워크 관리자들이 보다 쉽게 작업을 수행할 수 있도록 해 주는 솔루션이다[4]. OS/2 기반의 애플리케이션 관리 툴인 WSOD 1.0은 원격지 부팅 및 소프트웨어 설치가 가능한 애플리케이션 관리 툴로서, 서버 기반의 클라이언트 설정과 단순화된 애플리케이션 분산 기능을 제공하기 때문에, 기존의 PC를 NC로 이전하려는 업체에서 도입할 만 하다.

WSOD의 기본 동작을 살펴보면, 클라이언트에서 사용하게 될 애플리케이션과 운영체제를 서버에 설치함으로써, 각각의 클라이언트에서는 서버 측에 설치된 운영체제를 부팅할 수 있게 된다. 따라서 클라이언트에는 별도의 하드디스크를 장착하지 않아도 되는, 씬(thin) 클라이언트 환경을 제공하고, 도스, 썬 마이크로시스템의 자바, IBM, 윈도우 3.x 애플리케이션 등에 대한 중앙 집중화된 관리 기능을 제공한다[3].

WSOD는 애플리케이션 설치시 단순화된 처리 과정을 갖추고 있어서, 네트워크 상에서 새로운 소프트웨어의 설치 및 사용을 단순하게 하고 안정적으로 만들어준다. 그리고 서버에 설치된 애플리케이션은 WSOD의 관리하에 있는 어떤 데스크톱을 사용하더라도 동일한 환경에서 활용할 수 있다. 하지만 서버 상에 사용자 애플리케이션을 설치할 때 유의해야 할 사항도 있다. 상용 소프트웨어는 설치 시 많은 하위 디렉토리를 포함하게 되고, 보통 설치시 이러한 하위 디렉토리와 파일들의 패스가 문서화되어 있지 않기 때문에 그 파일들을 직접 찾고자 한다면 문제가 된다. 또 32비트 윈도우 애플리케이션을 지원하지 못한다는 단점을 지니고 있다. WSOD는 데스크톱 관리를 매우 단순화시키는 장점이 있지만, 하드웨어 및 소프트웨어 요구사항이 까다롭고 마이크로소프트 윈도우 95와 NT를 지원하지 못하는 단점도 있다. IBM은 이 문제를 해결하기 위해 써드 파티 솔루션을 제안하고 있다.

3. DL-NCS의 구현

3.1 DL-NCS의 구성

DL-NCS는 Windows NT에서 제공하는 기능을 최대한 활용하고, 본 논문에서 제안한 RPL 프로토콜을 이용하여 클라이언트가 서버측 하드디스크에 있는 MS-DOS, 윈도우 3.1/95/98 중 하나의 운영체제를 원격 부팅할 수 있게 구현하였다[4,5,6,7]. 본 논문

에서는 아직 대부분의 학원이나 중소기업체에서 많이 사용하고 있는 윈도우 95에 관련된 내용을 중심으로 설명한다.

DL-NCS는 Windows NT 서버와 2~50대까지의 PC 클라이언트가 이더넷으로 접속된다. 각각의 PC 클라이언트는 자체의 하드디스크가 없으며, 서버의 하드디스크에 설치된 윈도우 95를 부팅하게 된다. 먼저 DL-NCS의 구성을 살펴보면 그림 1과 같다. 서버는 펜티엄 166MHz 이상, RAM 64MB이상, 하드디스크 4.3GB 이상, Windows NT 4.0의 사양을 요구하고, 클라이언트는 486SX/DX 이상, RAM 8MB 이상, 부트 ROM이 장착된 LAN 카드를 필요로 한다. 그림 1에서 a1, a2 등으로 표현된 것은 서버에 등록된 각 클라이언트의 이름을 나타낸다.

Windows NT Server
DL-NCS Server
HUB
MS-DOS a1 | Windows 3.1 a2 | Windows 95 a3 | Windows 98 a4 | ··· | Windows 98 c9

그림 1. DL-NCS의 구성

3.1.1 공유 디스크의 사용

DL-NCS 클라이언트에서는 서버에 있는 하드디스크를 사용하게 됨으로 클라이언트 관점에서 보면, 하나의 네트워크 상에 있는, 하나의 드라이브를 공유하는 것과 유사하다.

모든 클라이언트가 서버측 하드디스크를 공유하기 때문에 2가지 부가적인 기술을 필요로 한다. 첫째, 서버는 모든 클라이언트에 대해 모든 디스크 동작을 지원할 수 있어야 한다. 따라서 DL-NCS 서버는 2~50 클라이언트를 지원하기 위해, 하드디스크가 4.3 GB 이상인 사양이 요구된다. 둘째, 클라이언트에 의해 적재되거나 실행될 모든 파일은 서버의 하드디스크로부터 읽혀지고 네트워크로 전송되기 때문에, 모든 클라이언트를 지원할 수 있도록 네트워크는 충분한 대역폭을 가져야 한다.

클라이언트가 자신의 하드디스크 드라이브를 갖고 있는 경우에도 DL-NCS를 구축할 수 있다. 대용량의 프로그램을 저장하기 위해, 또는 국부적인 교체(swap) 공간을 제공하기 위해 자신의 하드디스크 드라이브를 사용할 수 있다. 이 때 고려해야 할 기술적인 문제는, 대부분의 BIOS들이 네트워크 카드의 부트 ROM 대신에 하드디스크 드라이브로부터 부팅하려고 하기 때문에, 먼저 네트워크 카드로부터 부팅될 수 있도록 PC BIOS와 부트 ROM 간의 조화를 고려해야 한다.

3.1.2 네트워크 어댑터

DL-NCS 클라이언트가 부팅될 때 자신의 하드디스크에서 부트 명령을 찾는 대신, 네트워크 카드에 장착된 부트 ROM으로 제어가 넘겨지도록 설계한다. DL-NCS를 사용하기 위해 클라이언트는, 본 논문에서 개발한 부트 ROM 소켓이 장착된 네트워크 카드와 부트 ROM을 준비해야 한다. DL-NCS에서 윈도우 95 클라이언트를 원격 부팅하기 위해, 3Com EtherLink II, 3Com EtherLink III, AMD Series 2100 Ethernet, Intel EtherExpress 16, Intel EtherExpress Pro, Novell NE2000, Western Digital/SMC Ether Card Plus (8000 시리즈들)와 같은 ISA 버전의 네트워크 어댑터 카드들만이 클라이언트에 사용될 수 있다. Windows NT 서버가 네트워크를 통해 MS-DOS, 윈도우 3.1 또는 윈도우 9x 클라이언트들을 기동시킬 때 Remoteboot 서비스를 사용한다.

3.1.3 NetSetup 유틸리티

클라이언트에 하드디스크가 없어도 설정(setup)할 수 있도록 하기 위해, 윈도우 95A 버전에만 제공되는 netsetup.exe 프로그램이 필요하다. 이 프로그램이 설치된 서버 측에서 부팅 디스켓을 만든다. DL-NCS에서 클라이언트가 최초로 부팅을 하려고 할 경우, 이 부팅 디스켓을 이용하여 DL-NCS 환경에 맞게 설정할 수 있다.

3.2 DL-NCS 구축

3.2.1 Remoteboot 서비스 설정

DL-NCS의 클라이언트 PC에서 서버에 설치된 윈도우 95를 원격부팅할 수 있도록 하기 위해, 먼저 Windows NT에서 제공하는 Remoteboot 서비스를 설정해야 한다. 그 절차는, DLC와 NetBEUI 프로토

콜을 서버 상에 설치하고, Network Settings 대화상자에서 Remoteboot 서비스를 선택하며, 원격 부트 설치 대화상자에서 설치하고 싶은 원격 부트 디렉토리의 경로와 디렉토리 이름을 입력한다. 초기 값은 <systemroot>\Rpl이다.

3.2.2 RPL 프로파일 작성

DL-NCS에서 RPL 프로토콜이 동작되도록 하기 위해, 먼저 DL-NCS 서버측에 RPL 프로파일을 등록해야 한다. 이 기능은 rplcmd 유틸리티에 의해 이루어지는데, 이 유틸리티를 이용하면 워크스테이션(클라이언트 PC) 데이터베이스의 레코드를 조사하거나 갱신할 수 있다.

1) 새로운 어댑터 등록

새로운 네트워크 어댑터에 대한 부트 블록 레코드를 만들기 위해, rplcmd 유틸리티를 사용하여 어댑터와 공급자 ID에 대한 부트 블록 정보를 추가한다. 다음과 같이 윈도우 95에 대한 부트 블록 레코드를 만들 수 있다.

```
Adapter Boot Config Profile Service Vendor Wksta [Quit]:b
Add Del Enum : a
BootName=W95X
VendorName=00C026DC2B4D
BbcFile=BBLOCK\NETBEUI\dirname\W95BB.CNF
        All other parameters are optional
BootComment=Name of this adapter
WindowsSize=0

Adapter Boot Config Profile Service Vendor Wksta [Quit]:v
Add Del Enum : a
VendorName=00C026DC2B4D
BootComment=Name of this adapter
```

위의 예에서 VendorName은 어댑터의 물리주소를 의미한다. 위에서 본 바와 같이 부트 블록 레코드를 만들고, 이어서 구성 레코드를 만들면 Remoteboot 서비스와 Remoteboot 관리자가 새로운 어댑터를 인식하게 된다. Remoteboot 서비스가 클라이언트로부터 부팅 요청을 받으면, 클라이언트의 네트워크 어댑터 ID와 관련된 워크스테이션 레코드를 찾는다. 이것이 없다면 어댑터 레코드를 만들지만, 클라이언트를 부팅시키지는 않는다. 그리고 워크스테이션 레코드와 연관된 프로파일, 프로파일을 만드는데 사용될 구성 레코드를 결정한다.

2) 새로운 어댑터에 대한 구성파일 생성

특별한 네트워크 어댑터 타입을 가진 윈도우 95 클라이언트를 지원하기 위해서는 먼저 MS-DOS 6.2x 구성 파일과 윈도우 95 구성파일을 만들어야 한다. 본 논문에서는 MS-DOS 6.2x 구성 파일을 작성하는 방법에 대해서는 생략하고, 윈도우 95 구성파일을 만드는 방법 중 새 어댑터에 대한 부트 블록 레코드를 추가하는 방법에 대해서만 설명한다.

① C:\Winnt\Rpl\Bblock\Netbeui\adapter 디렉토리에서 Dosbb.cnf 파일을 W95bb.cnf 파일로 복사한다.
② 새로운 W95.cnf 파일에서,
EXE BBLOCK\I13.COM ....을
EXE BBLOCK\W95I13.COM .... 으로 수정한다.
③ rplcmd 유틸리티를 사용하여, 새 어댑터에 대한 부트 블록 레코드를 다음과 같은 방법으로 추가한다.

```
C:\Winnt\Rpl>rplcmd
Adapter Boot Config Profile Service Vendor Wksta [Quit]: b
Add Del Enum : a
BootName=W95X
VendorName=00C026DC2B4D
BbcFile=BBLOCK\NETBEUI\dirname\W95BB.CNF
        All other parameters are optional
BootComment=Name of this adapter
WindowsSize=0
```

④ rplcmd 유틸리티를 사용하여 새로운 구성 파일에 추가한다.

3.2.3 원격 부팅용 윈도우 95의 설치

DL-NCS에서 클라이언트가 윈도우 95를 원격 부팅할 수 있도록 하기 위해, 먼저 DL-NCS 서버의 C: 드라이브의 바탕화면, ['WWW'의 Machine(M:)] 드라이브, 그리고 ['WWW'의 Windows(S:)] 드라이브에 윈도우 95A 버전을 복사해둔다. 이어서 윈도우 95 setup 디스켓을 만들기 위해 3.2.4절을 수행하고, 이 디스켓을 이용하여 최초의 윈도우 95 클라이언트를 설치하며, 이어서 다른 클라이언트를 setup한다.

3.2.4 윈도우 95 setup용 디스켓 생성

각 클라이언트에서 윈도우 95를 원격 부팅할 수 있도록 하기 위해, DL-NCS 서버에서 setup용 디스켓을 만들어야 한다. 네트워크 클라이언트 관리자에서 네트워크 설치 시동 디스크 만들기를 선택하고, 대상 워크스테이션 구성창에서 네트워크 클라이언트(C:)는 윈도우 95로 하며, 네트워크 어댑터 카드(A:)는 NE2000 Compatible로 선택한다. 네트워크 프로토콜의 내용을 NetBEUI 프로토콜로 변경하여 setup용 디스켓을 만든 후, 다음과 같이 autoexec.bat와 config.sys 내용을 변경함으로써 윈도우 95용 setup 디스켓을 완성한다.

```
autoexec.bat :
  path=a:\net
  a:\net\net start
  net use s:\\www\windows
  net use m:\\www\machine
  s:\setup /t:m:\a1.tmp

config.sys :
  files=30
  device=a:\net\ifshlp.sys
  lastdrive=z
  device=a:\net\himem.sys
  device=a:\net\emm386.exe noems
  dos=high,umb
```

3.2.5 원격부팅을 위한 윈도우 95 setup

3.2.4절에서 생성한 setup 디스켓으로 a1 클라이언트에 윈도우 95를 setup하려면, CMOS 환경설정 화면에서 A:로 부팅 되도록 설정한 후, rset8019.exe가 있는 디스켓으로 부팅해서 LAN 카드에 대해 IRQ는 3, I/O 포트는 300으로 설정한다. 그리고 원격부팅용 setup 디스켓을 이용하여 다시 부팅한 후, 화면에 출력되는 지시에 따라 설정하면 하드디스크가 없는 클라이언트에서 윈도우 95를 부팅할 수 있게 환경을 설정할 수 있다.

이어서 다른 클라이언트에 윈도우 95를 setup하려면 다음 과정을 수행한다.

① A:\autoexec.bat의 내용중 's:\setup /t:m:\a1.tmp'에서 a1.tmp를 a2.tmp 등으로, 각각의 클라이언트 PC의 이름으로 변경한다

② A:\net\system.ini의 내용 중 computername=a1과 username=a1에서 a1을 각각의 클라이언트 PC의 이름으로 변경한다.

③ 마지막으로 C:\windows\machine.ini에서 LAN-card의 물리주소를 등록한다.

3.3 제안된 RPL(Remote Program Load) 프로토콜

3.3.1 Remoteboot 프로세스

Windows NT의 Remoteboot 서비스는 서버의 하드디스크에 있는 윈도우 프로그램을 부팅할 수 있게 한다. 이 때 각 클라이언트들은 RPL(Remote Initial Program Load) 프로토콜이 저장된 ROM 칩을 장착한 네트워크 어댑터(LAN-card)를 가지고 있어야 하며, 클라이언트가 부팅될 때 이것이 서버로부터 시작과 구성에 관한 소프트웨어를 가져오게 한다. 이로 인해 클라이언트는 하드디스크를 가지고 있지 않아도 된다.

3.3.2 RPL 프로토콜

본 논문에서 제안한 RPL 프로토콜은 기본적으로 그림 2와 같이 동작되며, 구체적으로 설명하면 다음과 같다.

1) DL-NCS의 클라이언트가 켜졌을 때, 네트워크 어댑터가 초기화되고 부트요청(boot request) 프레임이 서버로 브로드캐스트 된다. 그러면 Remoteboot 서비스를 제공하는 DL-NCS 서버가 클라이언트의 어댑터 ID가 들어 있는 부트요청 프레임을 수신한다.

2) Remoteboot 서비스는 자신이 가지고 있는 Remoteboot 데이터베이스를 점검하여, 이 어댑터

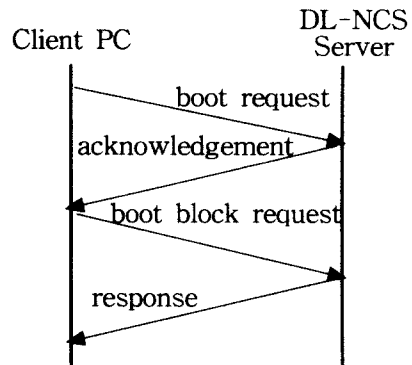



그림 2. 기본적인 원격부팅 동작

ID가 워크스테이션 레코드에 이미 등록되어 있는지를 점검한다. 들어 있지 않다면, Remoteboot 서비스는 이 어댑터 ID를 기록해 놓기는 하지만 클라이언트를 부팅시키지는 않는다. 클라이언트를 부팅 시키기 위해서는 관리자가 이 어댑터 ID 레코드를 Remoteboot 관리자를 사용해서 워크스테이션 레코드로 변환시켜 놓아야 한다. 워크스테이션 레코드에 이 어댑터 ID가 등록되어 있다면, Remoteboot 서비스 서버는 자신의 어댑터 ID가 들어 있는 FOUND 프레임을 클라이언트의 RPL ROM에게 보낸다(boot acknowledgment).

3) RPL ROM이 최초의 FOUND 프레임을 수신하면 SEND.FILE.REQUEST 프레임을, 가장 먼저 FOUND 프레임을 보낸 서버의 어댑터 ID로 보낸다(boot block request).

4) Remoteboot 서비스 서버가 SEND.FILE.REQUEST 프레임을 받으면, FILE.DATA.RESPONSE 프레임을 이용해서 부트 블록을 RPL ROM에게 보낸다. 이 때 원격 부트 데이터베이스에 있는 워크스테이션 레코드가 어느 부트 블록을 보내야 할지(MS-DOS용인지 윈도우용인지)를 지정한다. 부트 ROM이 마지막 FILE.DATA.RESPONSE 프레임을 받으면 부트 블록의 첫 부분으로 제어를 넘긴다. RPL ROM은 부트 블록에 의해 클라이언트에 적절하게 지정된 운영체제를 부팅 한다. 클라이언트가 MS-DOS나 윈도우 3.1을 운영체제로 한다면 여기서 원격 부팅 과정은 끝난다.

6) 윈도우 95를 운영체제로 삼고자 하는 클라이언트는 부팅 과정을 계속한다. 지금 이 단계에서 클라이언트는 DL-NCS 서버의 파일들을 이용해서 리얼모드(MS-DOS 7.0과 동일)로 운영되고 있다. 부팅을 마무리 짓고 윈도우 9x를 실행하기 위해서 클라이언트는 다음 과정을 수행한다.

① RAM 디스크를 만든다. RAM 디스크는 디스크 드라이브처럼 쓰려고 따로 마련해 둔 RAM 메모리 영역이다.

② 원격 부트 서버에서 윈도우 95의 리얼 모드 파일들을 RAM 디스크로 복사해 온다.

③ 윈도우 95 리얼 모드 네트워크 드라이버를 띄우고 DL-NCS 서버에 연결한다. DL-NCS 서버는 Windows NT 서버와 동일한 PC에 설정할 수도 있다. DL-NCS 서버는 Windows NT 서버에게 윈도우 95 파일들을 제공해서 클라이언트가 부팅 과정을 마칠 수 있도록 한다.

④ 해당 클라이언트가 기계 디렉토리(사용자 디렉토리)를 제공하는 DL-NCS 서버에 접속되게 한다.

4. 구현 결과 및 분석

자체의 하드디스크가 없는 클라이언트 PC에서 전원을 켜면 서버로부터 윈도우 95를 부팅하게 되었고, C: 드라이브는 Windows NT가 설치된 드라이브로서 클라이언트에서는 접근할 수 없도록 구현되었다. 그림 3은 클라이언트 PC가 부팅되었을 때 [내 컴퓨터]에 대한 화면을 보여준다. 이 그림에서 E:, M:, S: 드라이브는 서버에 위치한 하드디스크에 존재한다.

여기서 D: 드라이브는 [MS-ramdrive]로 지정되었으며, 이 드라이브에는 Winboot라는 디렉토리가 존재한다. 원격 부팅이 이루어질 때 서버의 ['WWW'의 Windows] 드라이브에 설치되어 있는, 윈도우 95의 시스템 프로그램을 서버로부터 복사하여 클라이언트의 RAM 영역에 저장하게 된다. 이 RAM 영역을 [MS-ramdrive]라고 부른다.

그림 4는 ['WWW'의 Machine(M:)] 드라이브의 내용을 보여준다. 원격부트 관리자에서 클라이언트 네트워크 어댑터 ID로 등록한 각각의 ID에 대해, 이 드라이브 상에 디렉토리를 배정해 주게 된다. 이 그림에서 a1~a9, b1~b9, c1~c9 등은 각각의 클라이언트에 배정된 디렉토리이며, a1 클라이언트는 a1 디렉토리에만 접근할 수 있고 다른 디렉토리에는 접근할 수 없도록 구현하였다. 따라서 각각의 디렉토리는 물리적으로는 서버에 존재하지만, 논리적으로는 로

그림 3. 클라이언트 PC상의 [내 컴퓨터] 내용

그림 4. 드라이브 M:의 내용

컬 하드디스크처럼 동작한다. 그림 4에서 보는 것처럼 일반 PC에서 개인 사용자가 사용하게 될 응용프로그램들을 이 드라이브에 설치하게 되고, DL-NCS에서는 여러 클라이언트가 이 드라이브에 설치된 응용 프로그램을 공유하게 된다. 이 화면에서 a1netlog와 같은 파일은 각 클라이언트에서 발생한 에러, 경로와 같은 정보를 갖는다.

본 연구에서 개발한 DL-NCS를 이미 상품화하여 여러 초중고등학교, 컴퓨터 학원, 중소 기업체 등에 설치하였다. DL-NCS의 성능을 평가하기 위해, 펜티엄 MMX233 MHz, RAM 64MB, 하드디스크 8.4GB 사양을 갖는 PC에 Windows NT를 설치하여 서버로 하고, 486DX66, RAM 16MB 사양을 갖는 25대의 PC를 클라이언트로 하여 DL-NCS를 구축하였다. 성능 평가를 위해 다음과 같은 2가지 환경에서 실험 하였다.

- 실험1의 환경
  - 사용한 허브: 더미 허브 (SURCOM HUB, 모델: Ethernet Perfect 532D)
  - 서버측 HDD 인터페이스: IDE방식
  - 서버측 LAN 인터페이스: 10BASE-T ISA 버스 이더넷 어댑터
  - 클라이언트측 LAN 인터페이스: 10BASE-T ISA 버스 이더넷 어댑터
- 실험2의 환경
  - 사용한 허브: 3COM 스위칭 허브 (모델: BaseLine-16462)
  - 서버측 HDD 인터페이스: SCSI 방식
  - 서버측 LAN 인터페이스: 100BASE-T PCI 버스 이더넷 어댑터
  - 클라이언트측 LAN 인터페이스: 10BASE-T ISA 버스 이더넷 어댑터

이와 같이 두 환경에서 실험한 결과를 표1과 2에 나타내었다. 표1과 2는 각각 실험1과 실험2의 환경에서 윈도우 95를 부팅하거나 응용 프로그램을 수행할 때 소요되는 시간을 초 단위로 나타내었다. 표1과 2에서 DL-NCS의 수행 시간 측정은 25대의 클라이언트 PC의 전원을 동시에 켰을 때, 가장 빨리 윈도우 95를 부팅하는데 소요된 시간을 [최소]로 측정하였고, 마지막으로 부팅된 클라이언트 PC에게 소요된 시간을 [최대]로 표현하였다. 또 (PC)로 표현된 항목은 독립된 PC에서 수행될 때 소요되는 시간을 의미한다.

그림 5는 서버측 하드디스크 인터페이스 방식, LAN 어댑터 종류, 허브의 종류에 따른 DL-NCS의 성능을 나타낸다. 즉 표 1과 표 2에서 측정된 결과를 비교한 것이다. 이 그림에서 범례에 사용된 실험 1, 실험 2는 각각 실험 1 환경과 실험 2 환경에서 측정한 값을 나타내고, stand alone은 독립된 PC에서 측정한 값을 나타낸다. 표 1을 보면, 실험 1의 환경에서 윈도우 95를 부팅하는데 소요된 시간은 최소 1분 18초, 최고 1분 57초이다. 표 2를 보면, 실험 2의 환경에서 윈도우 95를 부팅하는데 소요된 시간은 최소 1분 5초, 최대 1분 10초 정도라는 것을 확인하였다. 이것은

표 1. 실험1 환경에서의 DL-NCS 성능

실험 항목	수행시간(DL-NCS)		수행시간
	최소[초]	최대[초]	(PC)[초]
윈도우 95 부팅	78	117	65
한글 97	24	38	20
Photo Shop	141	190	120
MS PowerPoint	19	25	15
MS Excel	42	62	40
MS Access	18	29	15
Visual Basic	23	32	19
AUTO CAD 14	74	175	60

표 2. 실험2 환경에서의 DL-NCS 성능

실험 항목	소요시간(DL-NCS)		소요시간
	최소[초]	최대[초]	(PC)[초]
윈도우 95 부팅	65	70	65
한글 97	21	25	20
Photo Shop	123	137	120
MS PowerPoint	15	19	15
MS Excel	41	44	40
MS Access	16	19	15
Visual Basic	19	23	19
AUTO CAD 14	60	70	60

그림 5. 네트워크 장비에 따른 DL-NCS의 성능 비교

독립된 PC에서 윈도우 95를 부팅하는데 소요되는 시간이 1분 5초 정도임을 감안하면, 본 DL-NCS는 훌륭하게 동작함을 알 수 있다. 특히 실험2의 환경에서는 25대의 클라이언트를 동시에 사용해도 독립된 PC와 거의 같은 속도로 동작함을 알 수 있다.

또 표 1에서 보는 바와 같이, 실험 1 환경에서 한글 97을 동시에 실행시켰을 때 가장 빨리 혼글 화면이 뜬 것은 24초가 소요되었고, 가장 늦은 것은 38초가 소요되었다. 또 실험 2 환경에서 측정한 결과인 표 2를 보면, 혼글 화면을 띄우는데 최소 21초, 최대 25초가 소요되었다. 독립된 PC에서 20초 정도가 소요되므로, 실험2 환경으로 DL-NCS를 구축하면 대부분의 응용프로그램을 독립된 PC에서와 비슷한 속도로 사용할 수 있다는 것을 알 수 있다.

현재 본 DL-NCS를 설치한 사이트 중 가장 많은 클라이언트를 접속한 것은 486SX 50대로서, 실험1에서 보여준 사양으로 구축하였다. 이 시스템에서는 윈도우 95 부팅 속도가 최소 4분 50초, 최대 8분 55초 정도로 속도가 떨어졌지만, 응용 프로그램을 사용하는데는 큰 어려움이 없었다. 실험1의 환경으로 시스템을 구축하면 구축비용을 절감할 수 있고, 실험2의 환경으로 구축하면 좀 더 빠른 속도로 DL-NCS를 사용할 수 있다.

본 DL-NCS의 장점을 살펴보면 다음과 같다.

1) 새로 PC를 구입하여 네트워크를 구축하려고 하는 경우, 각각의 클라이언트에 하드디스크와 CD ROM드라이브를 장착하지 않아도 되므로 장비 도입비용을 줄일 수 있다.

2) 486SX/DX나 펜티엄 100MHz 이하급의 PC를 사용하여 DL-NCS를 구축하면, 성능이 낮은 PC에서 윈도우 95나 98과 같은 운영체제를 사용할 수 있고, 이 운영체제에서 사용 가능한 응용 프로그램을 이용할 수 있게 된다.

3) 서버에만 필요한 응용 프로그램을 설치해 두면, 모든 클라이언트에서 공유해 사용할 수 있으므로 관리하기가 쉽다. 본 시스템에서는 한글 97, AutoCAD 14, CADKEY, CADPOWER, HTT 타자연습, Visual C++, C/C++, Delphi, Excel, PowerPoint 등과 같은 응용 프로그램을 공유하여 사용할 수 있다.

5. 결론

본 논문에서는 펜티엄 166MHz 이상, RAM 64MB 이상, 하드디스크 4.3GB 이상인 PC에 Windows NT 4.0을 설치하여 DL-NCS 서버를 구축하고, 486SX/DX급 PC를 클라이언트로 하여 DL-NCS를 설계, 구현하였다. Windows NT의 기능을 최대한 활용하여 RPL 프로파일과 원격 부팅용 setup 디스켓 등을 생성하였다. 또 RPL 프로토콜을 제안하여 LAN 카드에 부트 ROM이란 이름으로 장착하였다. 이로써 각 클라이언트는 하드디스크가 없는 상태에서 윈도우 9x를 원격 부팅할 수 있고, 서버측 하드디스크에 저장된 응용프로그램들을 동시에 공유해서 사용할 수 있다.

성능 평가를 위해 25대의 클라이언트(486DX66)의 전원을 동시에 켜 본 결과, 훌륭하게 동작함을 확인하였다. 더미 허브를 사용하고, IDE방식의 하드디스크와 ISA 버스 이더넷 어댑터를 채택한 서버 환경에서 측정한 수행 시간은 독립된 PC에서의 수행시간보다 최고 2배 가까이 걸렸지만, 사용하는데는 어려움이 없었다. 스위칭 허브를 사용하고, 서버에 SCSI 방식의 하드디스크와 PCI 버스 이더넷 어댑터를 설치하면 수행속도가 크게 향상되어 독립된 PC에서와 비슷한 수행속도를 얻을 수 있었다. 전자의 환경으로 시스템을 구축하면 구축비용을 절감할 수 있고, 후자의 환경으로 구축하면 좀 더 빠른 속도로 DL-NCS를 사용할 수 있다.

본 DL-NCS를 설치하면, 하드디스크나 CD-ROM과 같은 주변 장치 설치비용을 절감할 수 있고, 486SX/DX와 같은 낮은 성능의 PC를 재활용할 수 있으며, 한 벌의 응용프로그램만 설치해도 여러 클라이언트가 공유할 수 있으므로 시스템 관리 및 소프트웨어 구입비용을 절감할 수 있다.

앞으로 클라이언트의 대수를 증가시켰을 때 수행속도를 향상시키기 위한 연구가 계속되어야 하며, 본

시스템을 인터넷에 접속하기 위한 연구가 필요하다.

참 고 문 헌

[1] Disposable PC, BYTE, pp.114-123, 1998. 3.

[2] 박은식, "NC와 NetPC", MicroSoftware, pp.224-235, 1997. 8.

[3] Rollin White and Carla Hanzlik, "WorkSpace On-Demand", http://www.scoug.com/ os24u/ 1997/scoug711.2.workspace.html, pp.1-8, Nov. 1997.

[4] IBM, WorkSpace on-Demand 1.0, LANTIMES, 1998. 3.

[5] Philippe, "Automated Object Design: The Client-Server Case", IEEE Computer Mag., Vol.29, No.2, pp.62-66, Feb. 1996.

[6] Mark Lomas and Bruce Christianson, "Remote Booting in a hostile world", IEEE Computer Mag. Vol.28, No.1, pp.50-54, Jan. 1995.

[7] 정연기, 이광진, "원격부팅 시스템", 특허청 특허 출원번호:10-1999-0027859, 1999. 7.

정 연 기

1982년 2월 영남대학교 전자공학과 졸업(공학사)

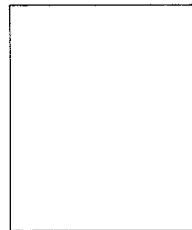
1984년 2월 영남대학교대학원 전자공학과 졸업(공학석사)

1996년 2월 영남대학교대학원 전자공학과 졸업(공학박사)

1985년 3월~1990년 2월 가톨릭상지대학 전산정보처리과 조교수

1990년 3월~현재 경일대학교 공과대학 컴퓨터공학과 부교수

1998년 1월~1998년 12월 호주 뉴캐슬대학교 전기 및 컴퓨터공학과 교환교수

관심분야 : 멀티미디어 통신, ATM/B-ISDN 기반의 초고속 정보 통신망, TMN/TINA 체계의 통신망 운용관리

이 광 진

1993년 2월 경일대학교 컴퓨터공학과 졸업

1997년 8월~현재 경일대학교 산업대학원 컴퓨터공학과 재학 중

관심분야 : 멀티미디어 시스템, 네트워크 컴퓨터